CHAPTER 1

Character Skeletons

Getting a character to move properly begins and ends with a good skeleton. It takes proper positioning and the right number of joints. The skeleton provides functionality and at the same time limits the ways geometry can deform. In this first chapter, you'll construct a skeleton and weight it to a modeled character. You'll then apply stock motion capture to test the character's deformations and the skeleton's validity.

In this chapter you will learn to:

- **Evaluate your characters geometry**
- **Paint weights**
- **Mirror and copy weights**
- **Apply motion capture**
- **Use the FBX file format**

Character Pipeline

The details of a character pipeline vary based on the character and change from company to company, and artist to artist, but for the most part, the overall process doesn't change. In this book, the pipeline is as follows:

PHOTO REAL PIPELINE

1. Establish the geometry.
2. Build a skeleton with basic functionality.
3. Add traditional skin weighting.
4. Create the facial anatomy.
5. Set up skin simulation.
6. Add facial hair using Fur.
7. Create and style hair.
8. Dress the character.
9. Implement secondary character effects.
10. Capturing a performance.
11. Add lighting and rendering.

Traditional skin weighting or binding only goes so far. The character in this book is fully clothed; therefore, his body doesn't need intense skin deformations. But the character needs to act. He needs to be able to speak and be convincingly expressive. To achieve this level of realism, it's crucial to match the real world as much as possible. Driving the character's face is an anatomically correct muscle system. The muscles squash and stretch and push and pull on the skin like real muscles.

Replicating human anatomy is half the battle. The muscles may move perfectly by themselves, but it's a different story when you need all of them to work together to form a smile or a frown. In comes *performance capture*. The term was coined by James Cameron while working on *Avatar*. Performance capture is the recording of every movement made by an actor. Well, maybe not every movement, but enough motion to identify speech, expressions, and of course full-body activities.

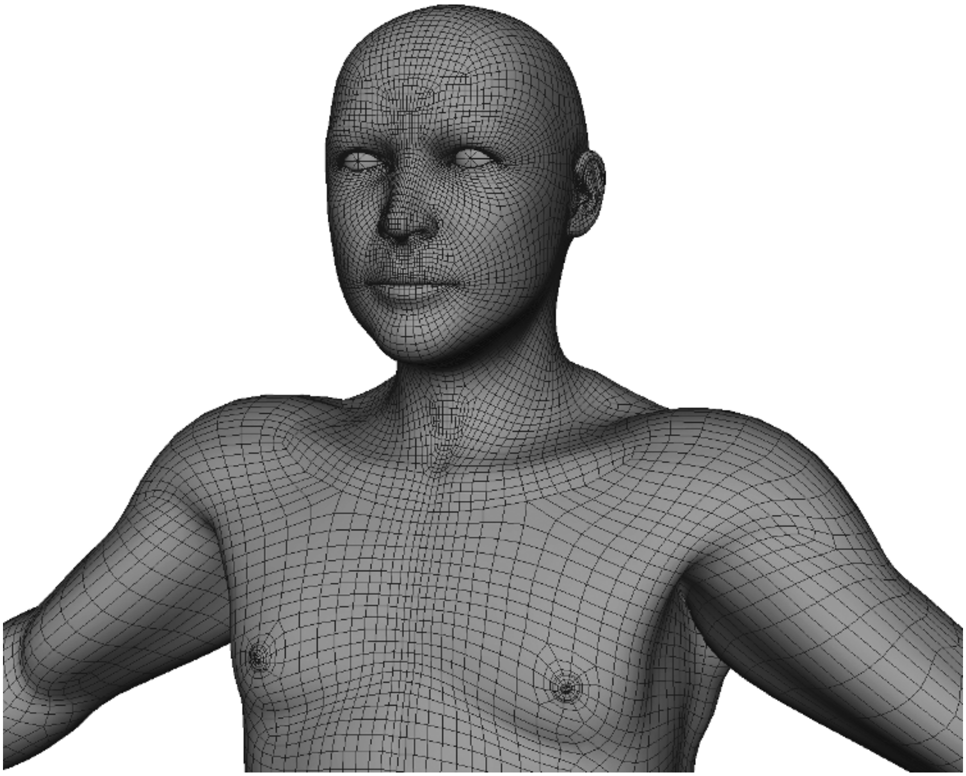
Skin Evaluation

Prior to adding joints, you should always evaluate your character's geometry or *skin*. It should have proper edge flow and enough faces for detailed deformations. Computer speed is constantly on the rise. Thanks to more RAM and accelerated GPUs, software has been able to increase production complexity. In the past, numerous techniques were used to push as much of the character's geometry as possible off until render time. This is still practiced today; but you can add more geometry to the character's actual bind. The

increase makes it possible to have deforming wrinkles and folds with accurate collision. This increase also demands a more intricate setup.

In this book, the character you'll create is clothed. Most of his skin is covered; only his hands and face are exposed. As a result, his body can have basic deformations, while his face is complex. Figure 1.1 shows the geometry of the character named Jack who is the focus of this book.

Figure 1.1
Jack's geometry



The geometry's density is consistent across the entire surface. Using predominantly quadrangle polygons funnels the edge flow from the body and arms up into the head. The head, which is significantly smaller than the overall body, benefits from compacting the incoming geometries into less space. Therefore, you have a good balance of geometry to support simulated clothing over the body and detailed deformations on the exposed face.

You use a muscle-based rig to drive the face. Blend shapes or morph targets are often used to animate facial expressions. However, they generate unnatural motion between each target and cause the development artist to generate hundreds to thousands of shapes to counteract the effects of the process. Muscle rigs make sense. Real muscles move in a linear fashion, and replicating this motion in computer graphics (CG) is simple. The

combined motions of groups of muscles create a facial expression. Tweaking this expression is a matter of moving a muscle a little up or a little down.

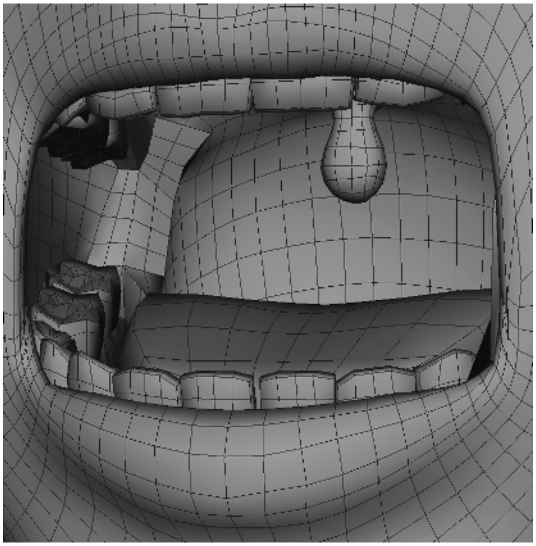


Figure 1.2
The interior of Jack's mouth

The geometry of the head is attached to the muscles. The muscles drive the geometric skin using nCloth for both the muscles and face geometry. This creates very complex deformations. It also makes it necessary to isolate the head from the rest of the body. You use basic weighting to control everything but the face. Detaching the head from the body allows the head to be nCloth. Ultimately, the head is a hybrid of nCloth and traditional weighting. The seam where the head and body connect is hidden by the character's clothes, but it's nevertheless held together by the basic weighting.

It's also important to mention the mouth. Having a character realistically express emotion and speak places a significant emphasis on the teeth, tongue, and rest of the mouth. Modeling these parts isn't as critical as modeling the exterior skin, but it's essential that they exist. Figure 1.2 shows the interior of Jack's mouth.

Base Skeleton

Building the skeleton is more of a technical achievement than an artistic one. The skeleton's functionality in the overall pipeline is primary. You must name the joints and orient them properly. You need to establish layers to toggle a joint's visibility as well as its selection status. These things are important; above all, however, is joint placement.

In the final setup, the character has two skeletons: a base skeleton and a motion-capture skeleton. The *base skeleton* is nothing more than a joint hierarchy that has been properly oriented. Its construction is simple but its joint placement is meticulously arranged.

Joints in the human body aren't always centered within the skin's volume. For instance, the bones along the spine are pushed to the back of the torso, with the ribcage protruding from the front. On a CG character, it's best to position the spine joints centered within the body; doing so makes the deformations function more naturally. The reason is that the human body is filled with bones, organs, and fluid, all of which play a factor in motion. A CG character is an empty volume, and its anatomy is implied. Figure 1.3 shows the CG character's spine placement.

Most often, what works best when placing joints is old-fashioned trial and error. You can bind the joints to the skin and test their rotation. Choosing Skin → Edit Smooth Skin → Move Skinned Joints Tool lets you make rapid changes to a joint's location. The tool works great for determining the position of the hip and knee (see Figure 1.4).

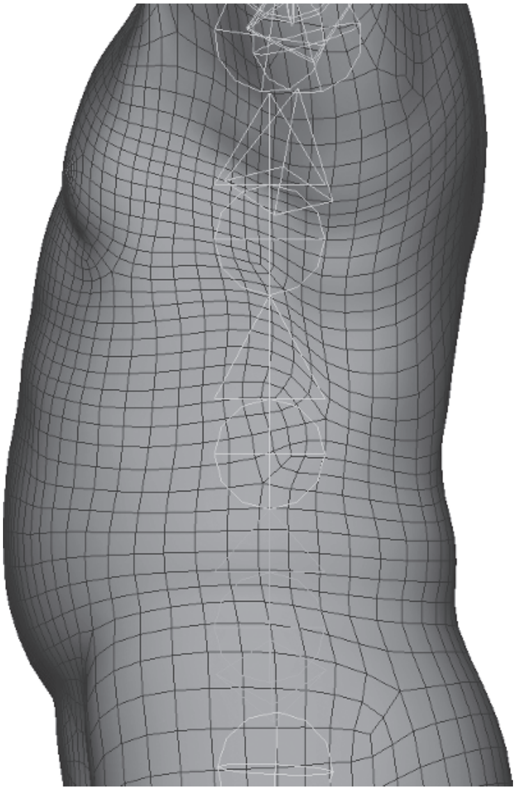


Figure 1.3
Jack's spine is positioned almost centered in his body and is vertically aligned.

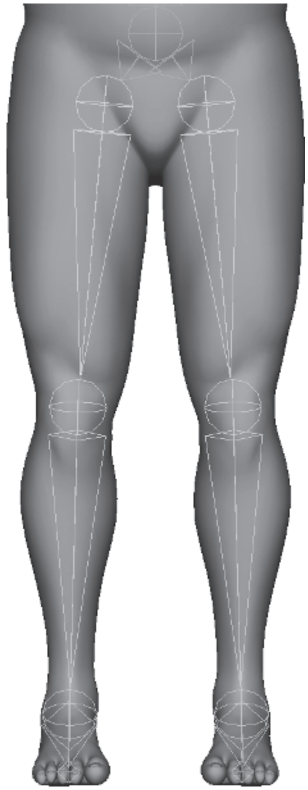


Figure 1.4
After testing several positions, the locations of the hip and knee are determined.

The arm joints are positioned in a similar fashion to the leg. However, their joints are located based on the shape of the character's skin. You want to pay close attention to the angle of the fingers to ensure proper rotation. Figure 1.5 shows the hand and the joint placement.

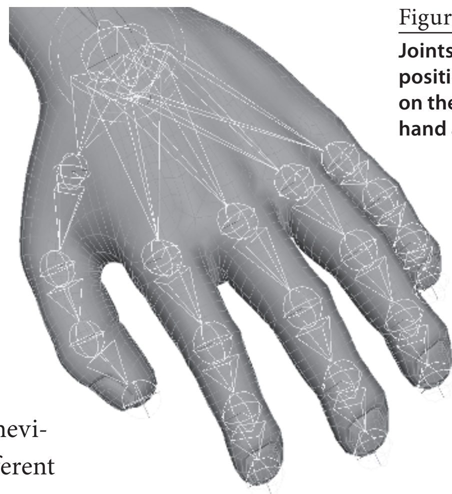


Figure 1.5
Joints are positioned based on the shape of the hand and fingers.

Performance Capture

Performance capturing is an important element in the photoreal character pipeline. Part of this process is creating a capture skeleton. You want the skeleton to match your character's skeleton as closely as possible. The more it deviates, the harder it is to transfer its motion. There will inevitably be some difference, but it's possible to calibrate to different skeletons.

To maintain the freedom to design and build any character, you need a system for calibrating the two rigs. The better your calibration process, the greater your freedom. The capture skeleton is based on the actor whose performance you're capturing. For this book, the capture skeletons are generated using Autodesk's MatchMover program (see Figure 1.6).

Figure 1.6
The interface for Autodesk MatchMover

Many other tracking applications are available that you can use in place of MatchMover. Autodesk's MatchMover program offers the benefit of compatibility. Because it's an Autodesk product, files can be exchanged between Maya and MatchMover easily.

After you import multiple angles of video or film footage of the same action into MatchMover, you track points that are identifiable in all the camera views. The points in the footage are typically markers that were manually placed for the sole purpose of tracking. MatchMover turns these markers into points in 3D space (see Figure 1.7).

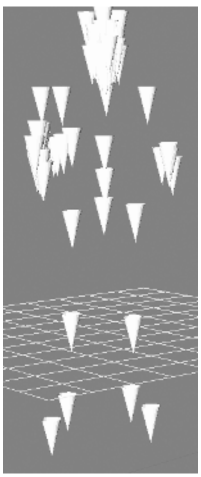


Figure 1.7
Markers from the imported footage turn into track points in 3D space.

The track points are then used to derive an animated skeleton by literally connecting the dots (Figure 1.8).

The track points are exported to Maya. In Maya, multiple skeletons are created and connected to each other. Scale plays an enormous role, no pun intended, in setting up both rigs. (Scale is addressed in detail in Chapter 7, "Performance Capture.")

The motion-capture skeleton and data aren't the critical part of the pipeline. The skeleton can change, and the data can change. It's what you do with the data that is decisive. With that said, you can't use any old data. For instance, there is a lot of downloadable

motion-capture data. Getting such data to work on your character is possible, as shown in the last two projects in this chapter, but it often fails for numerous reasons.

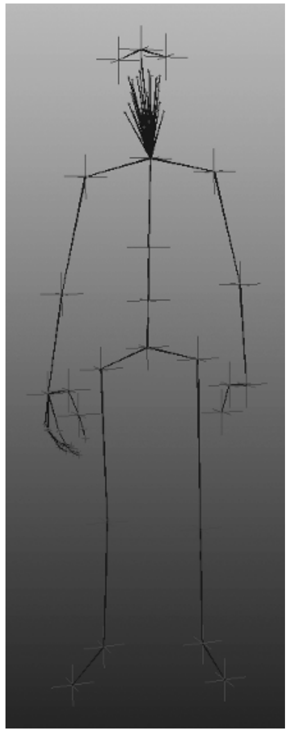


Figure 1.8
The track points are linked together to form a skeleton.

First, downloaded motion-capture data may fail due to unknown factors. You don't know how the data came to be. The actor's physique doesn't need to match your character, but knowing what the person looked like is helpful. Having a video of the actual performance is invaluable. Seeing the motion on joints in Maya isn't good enough: The data may be missing keys, or the actor's physique may be the opposite of your character. As mentioned earlier, it's possible to use differently proportioned skeletons, but not knowing why the data moves the way it does is a huge impediment. The rotation of a joint isn't a good representation of the joint's position. This is especially true for incomplete data sets like hands. Capturing finger motion is difficult, so the hand is sometimes a single joint or made up of a hand and a thumb joint. Regardless, the twisting motion of the wrist and upper arm is difficult and sometimes impossible to decipher without having a video reference of the actor. Remember, the skeleton must be calibrated first before the motion can be applied properly. If you can't align your character to the motion-capture skeleton, you've already lost the battle.

Another issue with motion-capture data is frame rate. Most capture sessions are done at 60 fps or higher to help in the tracking process. This is especially true for quick actions. A higher frame rate helps the software accurately track the data. It also helps when markers become obscured. Having more frames allows you to interpolate through the missing track points more smoothly.

The third problem with downloaded motion capture is the number of joints and marker placements. Motion capture is done for different reasons. The downloaded session may have been to get arm data only, so the lower half of the body was neglected. Again, having video of the capture session can reveal where the markers were placed. But videos may not reveal all them. Not having their exact locations makes it difficult and sometimes impossible to determine the differential between the motion-capture joint center and your CG character joint center.

All these reasons reinforce the need for your own personal motion-capture studio. Having your own setup gives you complete control over the data.

Weighting

Quaternion weighting helps skin deformations tremendously. You no longer need additional bones to help support the volume of skinned parts; you only need to add joints where your character needs them, making the process far more intuitive.

This also helps with the weighting process. Instead of the skin caving in at a bent joint, the volume is maintained, causing the skin to penetrate the opposing side. You can counteract the penetration by painting weight from the opposing side. You can even mix the quaternion weighting with classic linear weighting to have the skin lose some volume in the area, creating the illusion of collision. Both options create a more lifelike fold in the deformation. In the first project, Jack is weighted to his skeleton using quaternion weighting techniques.

Project: Painting Weights

You've evaluated the geometry and built the skeleton. Everything is in place for skin weighting. All of Jack's weights are done with quaternion weighting. The geometry to be weighted is one complete mesh. Ultimately, the mesh is divided into two pieces; but it's easier to weight a single mesh and then copy the weights to the final broken-up mesh. The following tutorial takes you through the entire process.

1. Open the `jackWeights_v01.ma` scene from the `chapter1\scenes` folder on the DVD. The scene contains the character's geometry as a single mesh and his skeleton. The geometry and skeleton have been assigned to respective layers.
2. Open the Outliner, and Ctrl+select the root_JNT node and the jack node. Choose Skin → Bind Skin → Smooth Bind tool options.
3. In the default settings, change Skinning Method to Dual Quaternion. Use Figure 1.9 to confirm your settings. Click Bind Skin to accept the options.

   The default bind does a decent job of weighting the character. It's far from perfect, of course, but it provides you with a clear path. Most of the needed corrections center on the torso. Figure 1.10 shows the torso's deformations.

4. Select clavicleLft_JNT, and rotate it to –15.0 in the *Z*. Select shoulderLft_JNT, and rotate it to 45.0 in the *Z*.

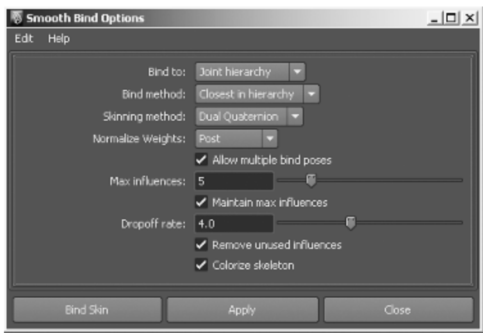


Figure 1.9
Change the Smooth Bind options.

CLAVICLE ROTATION

In the final rig, the clavicle will rotate only a little in the *Z* axis. To achieve a proper shoulder shrug, most of its motion will come from the joint's local X translation. Rotations are being used here simply for convenience. All the rotate values are at 0.0, making it easy to return the joint to its original position. Otherwise, you'd keyframe the joint's translation to retain its original value.

5. The spine and left shoulder joints are the first to tackle. Getting these deformations along the spine correct makes it easier to add the appendages. Select the skin, and choose Skin → Edit Smooth Skin → Paint Skin Weights Tool.
6. Knowing the target area to fix lets you customize the Paint Weights tool. Ctrl+select the following joints:
   - root_JNT
   - spine1_JNT
   - spine2_JNT
   - clavicleLft_JNT
   - shoulderLft_JNT
   - elbowLft_JNT
   - radiusLft_JNT
   - collar_JNT

   Choose the tack icon in the upper-right corner of the Influences list. The selected joints are now isolated.
7. Open the Gradient section, and select Use Color Map. The weight maps go from black and white to color. Select clavicleLft_JNT. Toggle the check box for Use Color Map a few times, and notice the difference in the display. The information is identical for both, but your eyes can't perceive the subtleties in the grayscale shading. Figure 1.11 shows the difference between the default weight maps and the colored ones.

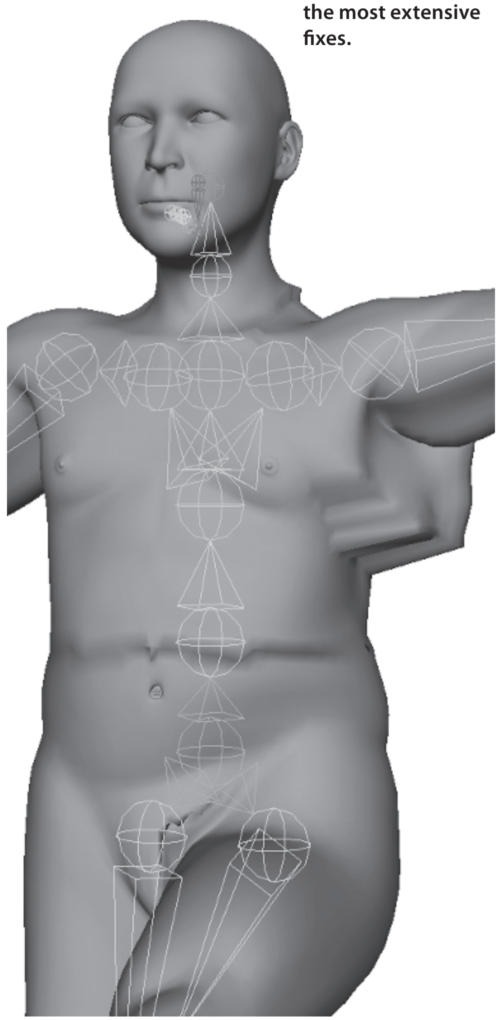


Figure 1.10
The torso requires the most extensive fixes.

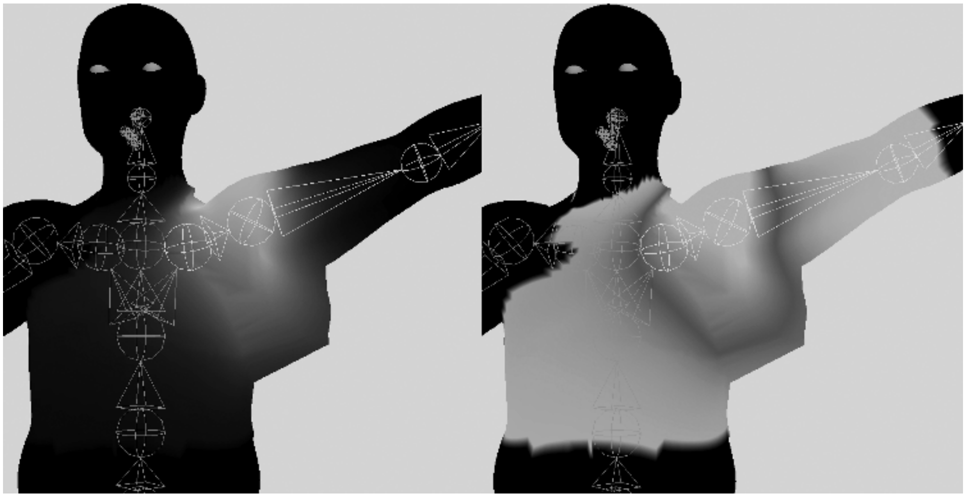


Figure 1.11
The black-and-white based weights maps are on the left, and the colored heat maps are on the right.

GRADIENT WEIGHT MAPS

There are two ways to display a joint's influence: grayscale shading and color ramps (also known as a *heat map*). The grayscale shading is straightforward: black is zero, and white is one. Heat maps increase the amount of visible contrast by displaying weight values as colored intensities. They facilitate the ability to paint without knowing the actual value of the weight. In other words, you can paint with greater accuracy. You can also see variants in the weighting that wouldn't normally be noticeable in traditional grayscale maps, such as values in the hundredths.

8. The first objective in the weighting process is to remove all the unnecessary weights. You could begin by choosing Skin → Edit Smooth Skin → Prune Small Weights, but this can also remove some necessary weighting. Instead, using the heat map, you can easily paint out what you don't need.

   Select radiusLft_JNT from the Influences list. The radius shouldn't have any effect on the skin past the elbow. Using a Replace value of 0, paint out all the blue weights up to the elbow.
9. Select elbowLft_JNT from the Influences list, and remove all of its blue values up to the elbow as you did for radiusLft_JNT in step 8.
10. Do clavicleLft_JNT next. Remove all the weights except a large area around the upper shoulder. Use your own body as reference. Place your hand over your heart, and then raise your shoulder. Feel the skin move? Slide your hand down and around the bottom of your ribcage. Keep raising your shoulder to feel how far the effects of the clavicle go. Figure 1.12 shows the remaining weight values.

Don't worry about getting the weights perfect or taking away too much. The weights are far from being finished, and it's much easier to come back with the Smooth Paint operation to fine-tune the results.

11. Spine2_JNT is a little out of hand. Its influence spreads across most of the body. Instead of trying to paint this out, it's easier to flood the weight to 0 and paint it from scratch. Change Paint Operation to Replace, set Value to 0.0, and click Flood. All of its weights are removed.
12. Change Value to 1.0, and change Brush Profile to the Solid brush. Paint from the bottom of the breast to just under the collar bone. Go from the center of the chest to the center of the back. Use the edge flow of the geometry as a guide. There is no need to paint the right side because you'll eventually mirror the weights. Use Figure 1.13 as a reference.

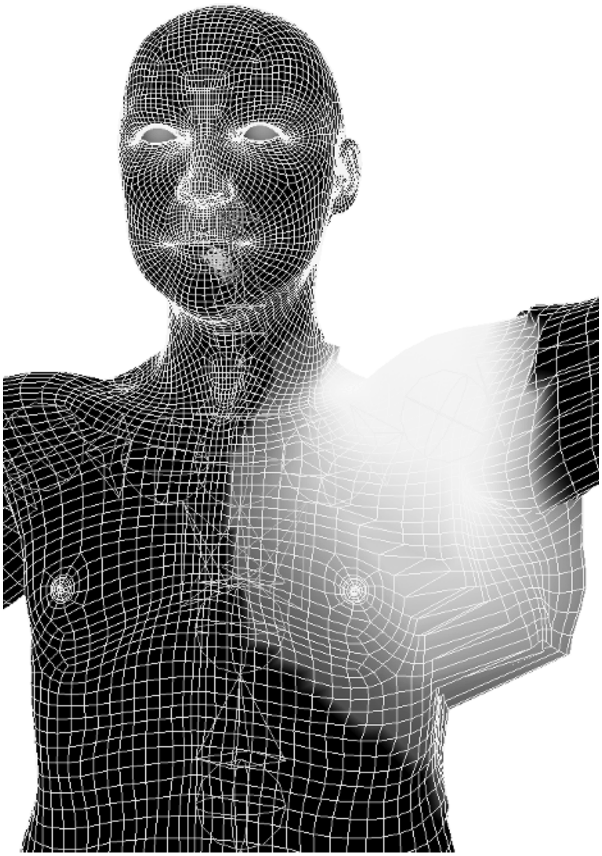


Figure 1.12
The weights for clavicleLft_JNT

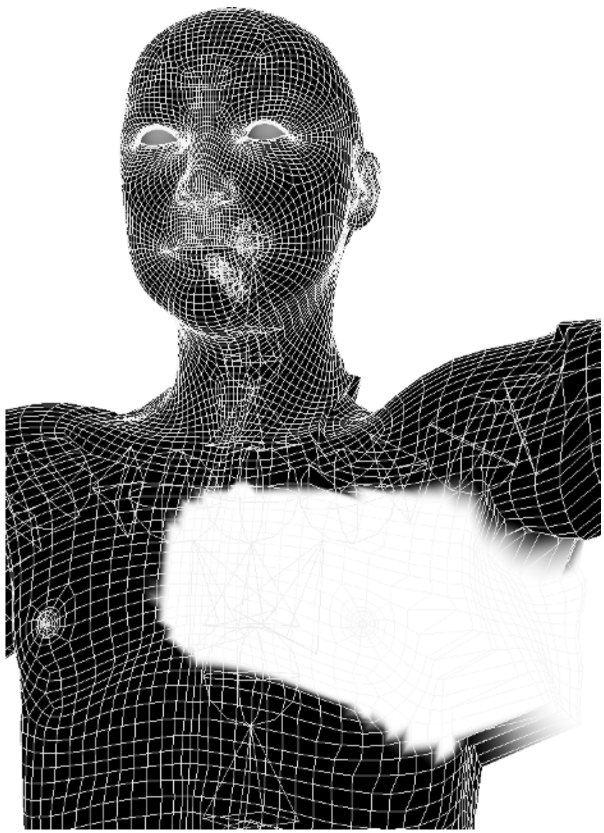


Figure 1.13
The weights for spine2_JNT

13. Switch the Paint operation to Smooth, and click the Flood button six times.
14. Choose shoulderLft_JNT. Change the Paint operation to Replace, and use a Value of 0.0. Paint out all the weights up to the shoulder. Use Figure 1.14 as a guide. Remember to do the back of the character as well.
15. You should start to see some progress. At this point, you can use the Smooth operation to refine the weighting. Keeping shoulderLft_JNT selected, smooth the weights around the shoulder area. You should end up with weighting similar to Figure 1.15.
16. Go through all the other joints, and continue to smooth the weights around the shoulder. The shoulder will look pretty good when you're finished. This is a good time to save your scene.

    To check your work so far, you can compare it to `jackWeights_v02.ma` on the DVD.
17. With the shoulder basically finished, let's move down the arm. Select elbowLft_JNT, and rotate it –30.0 units in the *Y*.

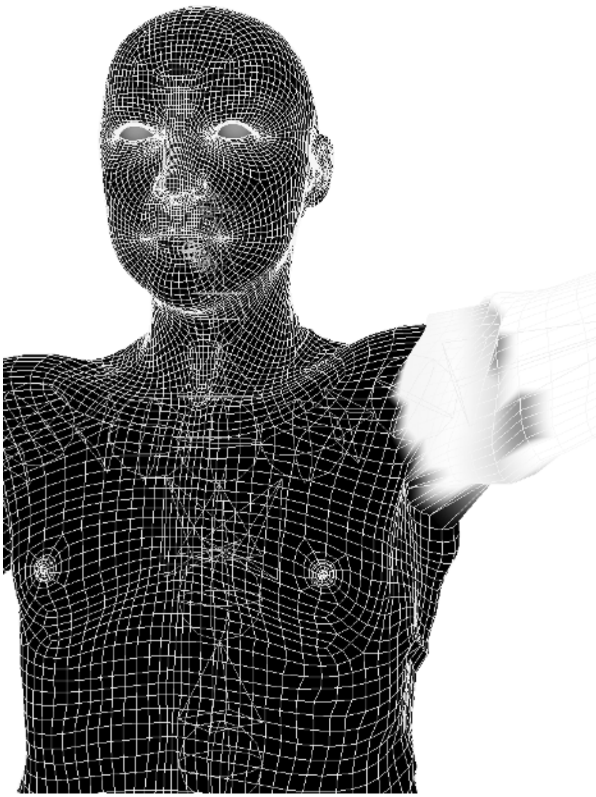


Figure 1.14
The weights for shoulderLft_JNT

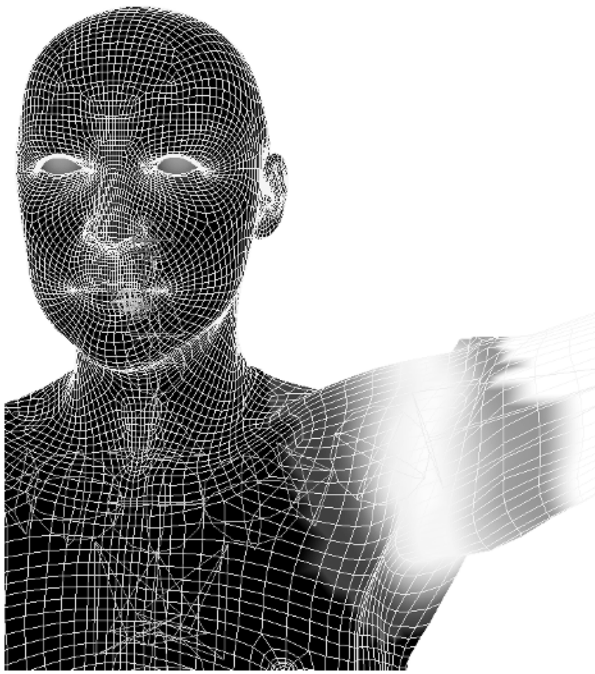


Figure 1.15
Smooth the weights for shoulderLft_JNT.

18. Select the character's geometry, and open the Paint Weights tool options. Select the following joints from the Influences list, and pin the selection:
    - clavicleLft_JNT
    - shoulderLft_JNT
    - elbowLft_JNT
    - radiusLft_JNT
    - wristLft_JNT
19. Click shoulderLft_JNT. Replace any weighting with 0.0 below the elbow.
20. Switch to elbowLft_JNT, and replace all the weights above the elbow with 0.0. Repeat for radiusLft_JNT.
21. Switch to wristLft_JNT, and replace all the weights with 0.0 down to the wrist joint itself.
22. Select the Smooth Paint operation. Toggle back and forth between the elbow and shoulder joints, smoothing the weights until they're blended nicely. When you're satisfied, clean up the smaller weights by painting them with a 0.0 value. The weights of the two joints should only bleed over the elbow area by a couple of edge loops. Figure 1.16 shows the finished results.
23. The radius and wrist joints work together; the radius rotates in the *X* while the wrist handles the *Y* and *Z* axes. The radius weights are good, except where they influence the wrist. Follow the same procedures you used for the previous joints to remove the unnecessary weighting.
24. Rotate wristLft_JNT to 40.0 units in the *Z*.
25. After you smooth the weights around the wrist joint, use the Scale Paint operation on both wristLft_JNT and radiusLft_JNT to reduce the weights and create a tight crease in the skin. A Value of 0.9 works well. Figure 1.17 shows how the wrist and the wristLft_JNT weight map should look.

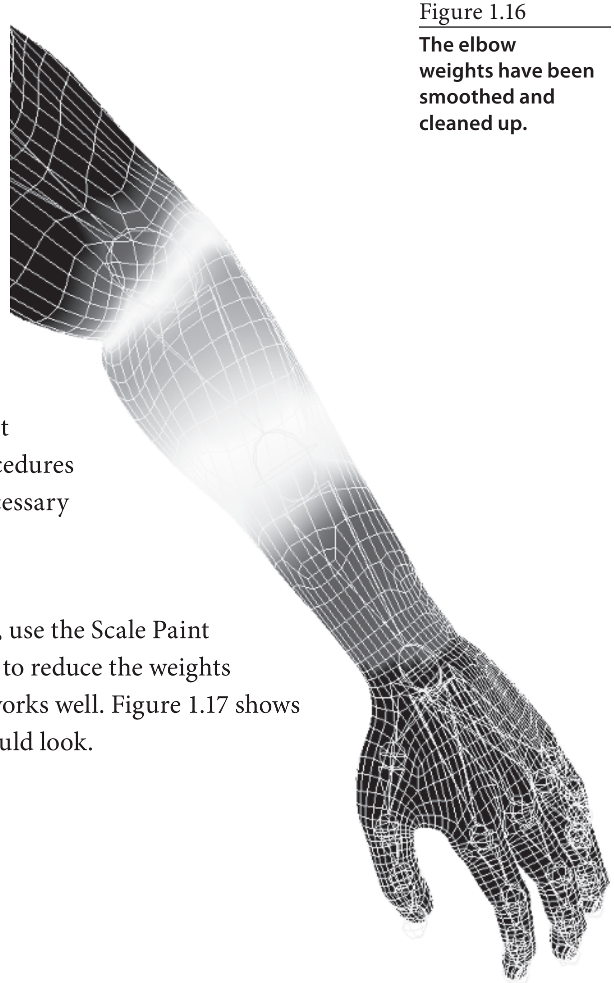


Figure 1.16

The elbow weights have been smoothed and cleaned up.

Figure 1.17
Scale the weights to get a crease around the bent wrist.

To check your work so far, you can compare it to `jackWeights_v03.ma` on the DVD.

26. Do the hip, leg, and neck using the same procedure you've followed for the other joints in this project. You can try them on your own or continue to the next project.

Mirror and Copying Weights

Symmetrical characters are great because you only have to build half of them. You can mirror just about every aspect. Skin weighting is no exception; mirroring significantly reduces the amount of redundant work.

Perhaps an even greater tool than mirroring is the Copy Skin Weights tool. It allows you to copy weights from one skin to another, regardless of the skeleton. It works on entire meshes as well as components. Copy Skin Weights is very flexible and a time saver.

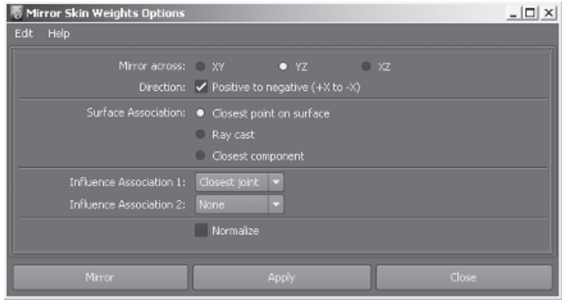
Project: Mirror and Copy Weights

Half of the character is weighted. In the next project, you'll mirror the weights and then clean them up. In the latter half of the project, you'll copy the single mesh weights over to the divided mesh with the head separated from the body. Follow these steps:

1. Open the `jackWeights_v04.ma` scene from the `chapter1\scenes` folder on the DVD. The scene picks up where the previous project left off. Half of the character is weighted.

2. Select Jack's geometry. Choose Skin → Edit Smooth Skin → Mirror Skin Weights. Use Figure 1.18 to set the mirror options. Click Mirror to apply the settings.

Figure 1.18
Mirror Skin Weight options



3. The Mirror Skin Weights tool is highly effective, but you can still have a few stray vertices. Rotate shoulderRght_JNT to 50.0 units in the Z. A single vertex from the torso is attached to the upper arm (see Figure 1.19).

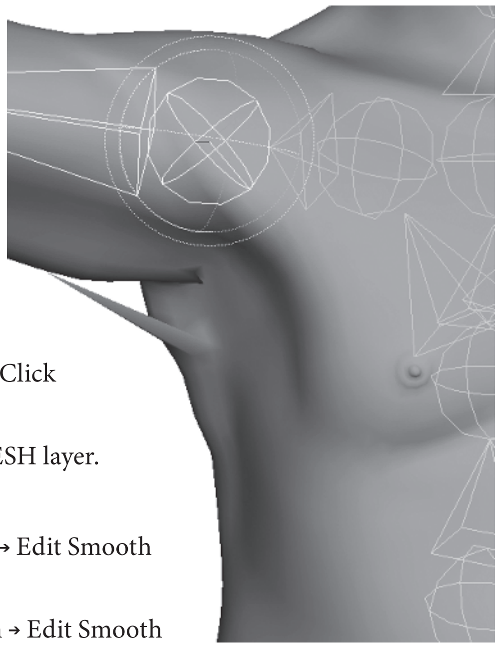
   Stray vertices like these are easy fixes. Select the problem vertex, and choose Skin → Edit Smooth Skin → Weight Hammer. In an instant, the vertex is averaged with the surrounding vertices and deforms naturally.

4. A closer inspection reveals a second vertex just above the armpit. Select it, and use Weight Hammer to fix it. Rotate the rest of the joints to make sure they all look good.

5. The weights are finished; you can now transfer them to the separated head and body geometries. Choose File → Import, and open the `jackParts.ma` scene file from the `chapter1\scenes` folder on the DVD.

   The head and body nodes are now in the scene. The two new pieces are assigned to the layer named PARTS.

Figure 1.19
One vertex under the arm didn't mirror properly.



6. Turn off the visibility for the JACK_MESH layer.

7. Shift+select root_JNT, head, and body. Choose Skin → Bind Skin → Smooth Bind. In the tool options, set Skinning Method to Dual Quaternion. Click Bind Skin.

8. Turn on the visibility for the JACK_MESH layer.

9. Shift+select jack and body.

10. Using the default settings, choose Skin → Edit Smooth Skin → Copy Skin Weights.

11. Shift+select jack and head. Choose Skin → Edit Smooth Skin → Copy Skin Weights.

    The weights successfully copy over to the two separate pieces as if they were weighted together. You can now delete the jack node.

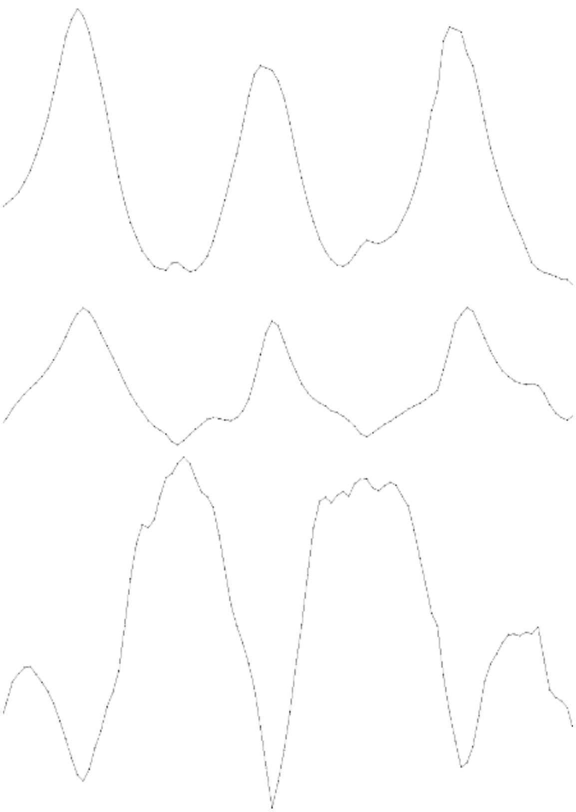
    To check your work so far, you can compare it to `jackWeights_v06.ma` on the DVD.

Motion Capture

Even the best animated characters still fall short of captured motion. You just can't beat the real thing. There are reasons why, but current hand-keyed animation isn't as convincing. Movie producers like to say it's because animators make characters do things that aren't natural. Animators say they needed more time or better direction. The truth is, the problem lies in the intricacies of the human body and the inadequacies of the current tools. To put it another way, it's bad timing.

When people and animals move, each bone is influenced by the one before it. This of course is *inverse kinematics* (IK). However, when one bone moves, the others counteract. Imagine typing on a keyboard. Your palms are resting on the desk, and it seems as though your fingers are doing all the work with your wrist keeping things stabilized. In order for your wrist to keep your fingers on the key board, your elbow shifts periodically. To support that elbow motion, your shoulder compensates. It doesn't stop there: every time you peck at a different key, different muscles are called into action. You can feel a similar effect by trying to stand still. All of your muscles fire to maintain your position. A single muscle applying force to a bone causes a chain reaction of hundreds of micromovements. Some go unnoticed, but when they're missing, we instantly recognize that something is wrong.

Figure 1.20
Three axes from motion-captured data



It's extremely difficult to calculate all those small motions and apply them in an animation. However, that's not the hard part. The real trick is finding the timing and patterns that exist among those motions. Look at Figure 1.20: it shows an excerpt of a motion-captured actor doing jumping jacks.

The top curve is the *Y* axis of the wrist. The middle curve is the *X* axis from the radius, and the bottom is the *X* axis of the humerus bone. Jumping jacks produce strong deviations in all the curves, but pay attention to how each curve has a relationship to the other. Even though they share the same peaks and values, how they get there isn't the same. Jitters and hiccups along the way indicate one bone supporting the other. You can see how the motion becomes seemingly more erratic as you go from the larger parent bone on top to the smaller child bone on the bottom.

Someday, computer-generated rigs will have this motion built into them, allowing animators to choose key positions and synthetic anatomy to fill in the rest. We're not far from having this technology. Maya has a precursor with full body IK (FBIK) rigs and human IK (HIK)

systems. Until these are capable of delivering a convincing human performance, we still need to rely on motion capture for the ultimate in photorealism.

As we mentioned at the beginning of the chapter, it's important to know where your motion-capture data comes from. You can use material downloaded from the Internet, but it often requires a lot of work and in the end produces less than adequate results. To use motion capture to its fullest potential, you must have complete control over every aspect, from the actor to the data that the motion capture outputs. With that said, more and more websites offer motion-capture data. Using this data can save you a tremendous amount of work, and the data is relatively cheap if not free.

Project: Motion Test

The Jack character is weighted, but you don't know how he'll perform. To test his abilities, you can apply downloaded motion-capture data to his skeleton. The data used in this project is from Carnegie Mellon University's Graphics Lab. They have a larger motion-capture database that you can access at `http://mocap.cs.cmu.edu/`. In addition, they have numerous other resources like converters and documentation. Follow these steps:

1. Open the `mocap_v01.ma` scene from the `chapter1\scenes` folder on the DVD. The scene contains the character's geometry weighted to his skeleton. The geometry and skeleton have been assigned to respective layers.

   A second skeleton is in the scene: it's significantly larger than Jack's skeleton and has captured motion of an exercise routine. It's assigned to a layer called MOTION_CAPTURE.

2. Open the Outliner, and select the root node. Group this to an empty group node by pressing Ctrl+G. Rename the group1 node **mocap_GRP**.

3. Select mocap_GRP, and scale it uniformly to the approximate size of Jack; 0.03605 is good (see Figure 1.21).

   Make a note of the scale you end up using; it can be used for all the motion capture done with the same actor. Later in the chapter, you'll apply different motion from the same actor.

4. Turn off the visibility on the body layer.

5. The group node is mainly needed for scale. You can also use it to offset the skeleton. You want to match the root joint of the motion-capture skeleton to Jack's skeleton. Before you can do this, you need to make sure you're at the default skeletal pose for both skeletons. In the Outliner, select the root node on the motion-capture skeleton. The timeline should fill up with red keyframes. Change the start time to –100. Go to frame –100, and then click the Step Forward One Key button on the timeline controls. Doing so reveals that the first keyframe is at –13. Change the timeline to match the beginning and end.

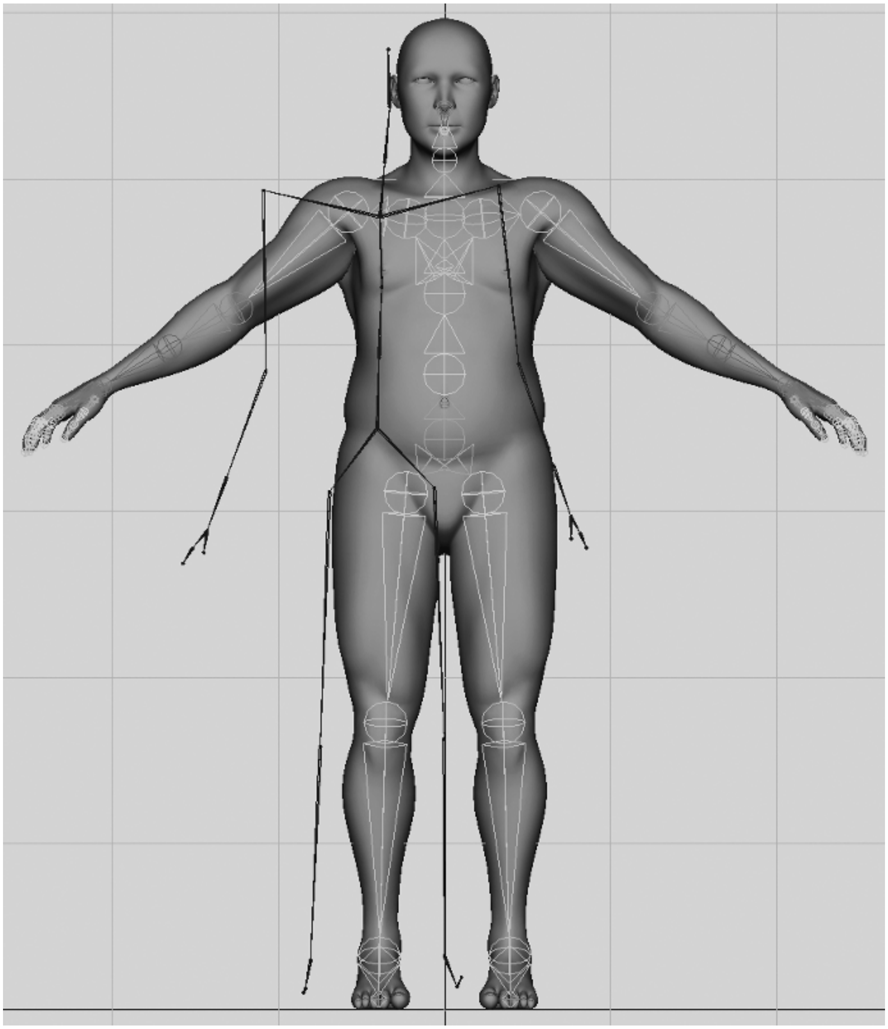


Figure 1.21
Scale the motion-capture group to match the size of Jack.

GROUP SCALE

At this point, you may want to reevaluate the character's scale. It should still be close enough. It's difficult to state whether you should scale the character in its default pose or its first frame of animation. If the first frame is a crouching or lying-down position, then you have to go with the default skeletal pose.

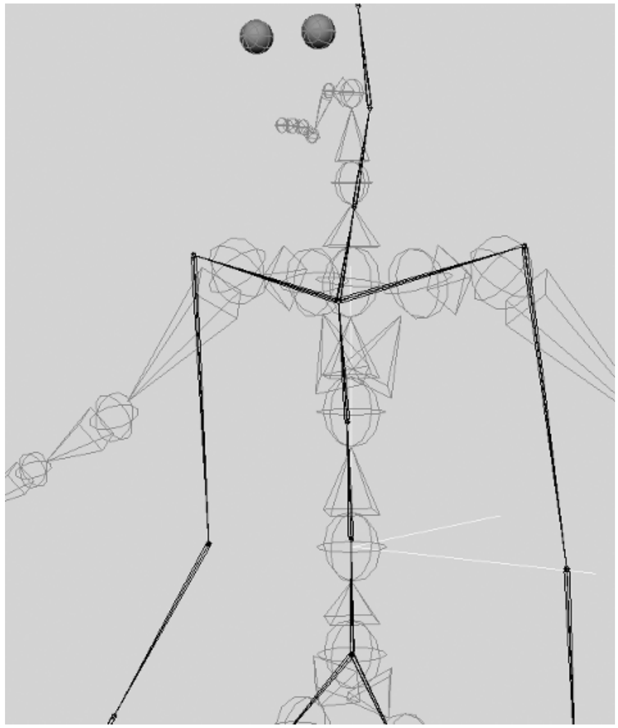
You don't want to go immediately to the default skeletal pose, because it isn't a true representation of the actor—it's rigid and unnatural. If you have control over the motion-capture session, it's ideal to have the actor pose in the same position as your CG character to obtain a suitable default skeletal pose.

6. Select mocap_GRP, and go to frame –13. Turn on Snap To Point. Press D, and snap the group's center to the root node of the motion-capture skeleton. With Snap To Point still on, snap mocap_GRP to Jack's root_JNT. The two skeletons are now collocated.
7. To get the skeletons to stay collocated, you need to add a point constraint. Shift+select root and root_JNT, and choose Constrain → Point. Reset the settings, and choose Add.

    Orient constraints would be the logical choice to align Jack's skeleton with the motion-capture skeleton, but the likelihood of the motion-capture skeleton being oriented with your skeleton is slim to none. The logical step is to apply the orient constraint and maintain its offset. You now have the tedious task of trying to align the joints manually.

    A better solution is to use single chain (SC) IK handles. You can snap the IK handle to the end of the bone you need to align with. Adding a point constraint forces the handle to follow along. An added benefit of the SC handle is that you can also constrain its orientation, allowing you to replicate a bone's twisting motion.

Figure 1.22
Add an IK handle to root_JNT.



8. The best place to start is at the root. Turn on Reference for the MOTION_CAPTURE layer.
9. Choose Skeleton → IK Handle Tool. Click Reset Tool at the top of the toolbar to return the settings to their defaults. Current Solver is set to ikSCsolver.
10. In Perspective, choose root_JNT and then spine1_JNT. Figure 1.22 shows the results.
11. Turn off Reference for the MOTION_CAPTURE layer, and turn on Reference for the SKELETON layer.
12. Shift+select upperback and ikHandle1. Choose Constrain → Point. Reset the tool to use the defaults, and click Add.
13. Select lowerback and ikHandle1, and choose Constrain → Orient. Make sure Maintain Offset is selected, and use the settings from Figure 1.23.

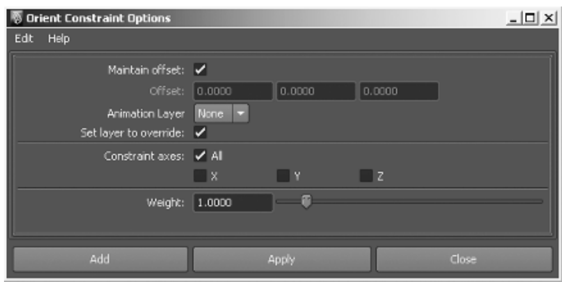
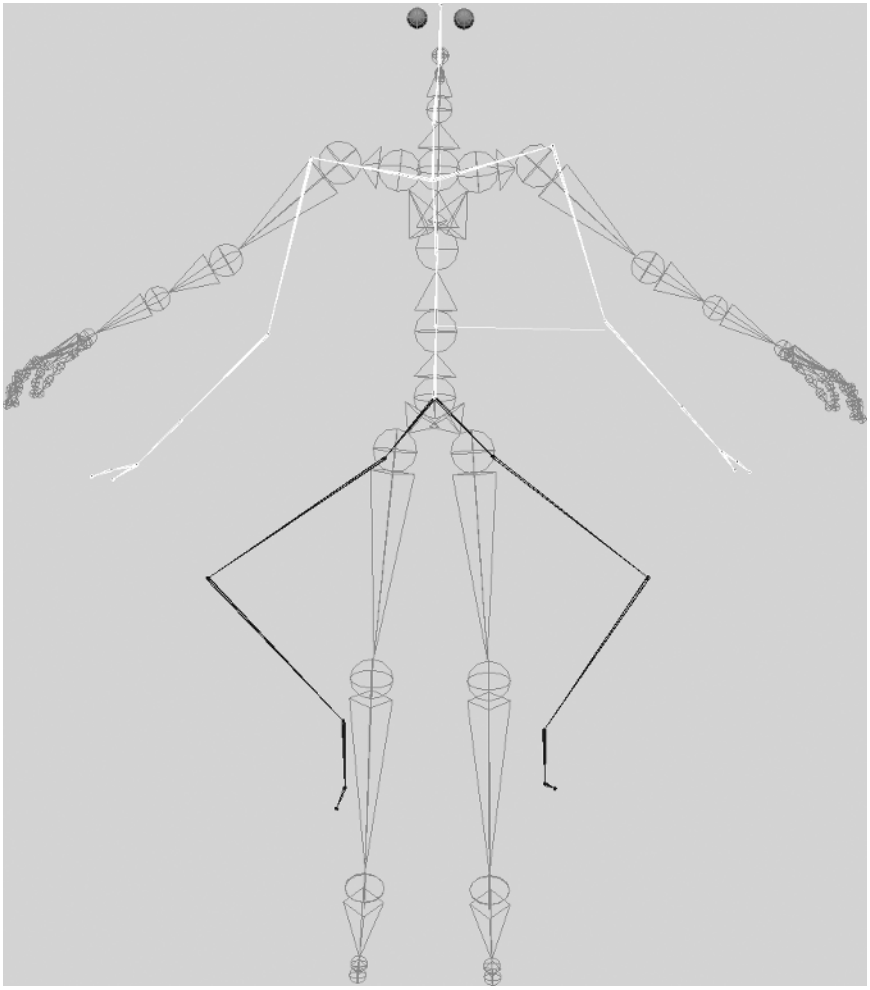


Figure 1.23
Orient constraint options

Click Play to see the constraints working. Figure 1.24 shows that the two skeleton roots are nicely aligned.

Figure 1.24
The captured character squats down with Jack's root node connected and perfectly aligned.



To check your work so far, you can compare it to `mocap_v02.ma` on the DVD.

Let's take a moment to review what you've just done. You point-constrained Jack's root node to the motion-capture root node. You only point-constrained the root node; none of the other joints are directly constrained. Then, you created an SC IK handle from Jack's root to his first spine joint. You point-constrained the SC handle to the motion capture's second spine joint, upperback. Doing this locked the *Y* and *Z* rotation of Jack's root joint to the pitch and yaw of the motion-capture skeleton. Next, you applied an orient constraint to the SC handle, effectively locking the *X* rotation to the twist of the motion-capture skeleton. The formula is to point-constrain the SC handle to the top of the bone and orient the handle to the root of the bone. Figure 1.25 illustrates.

14. Return to frame –13. You can now begin adding handles and constraints to the rest of the joints. For quick reference, Figure 1.26 shows the point and orient constraint options.
15. Add an SC handle from spine1_JNT to spine2_jnt.
16. Shift+select thorax from the motion-capture skeleton and ikHandle2. Add a point constraint.
17. Shift+select upperback from the motion-capture skeleton and ikHandle2. Add an orient constraint.
18. To finish the spine, add an SC handle from spine2_JNT and collar_JNT.
19. Shift+select lowerneck from the motion-capture skeleton and ikHandle3. Add a point constraint.
20. Shift+select thorax from the motion-capture skeleton and ikHandle3. Add an orient constraint.
21. For the left leg, follow the same procedure all the way down the skeletal chain. Here is a list of the constraint mappings:

SC IK Handle	Point Constraint	Orient Constraint
ikHandle 4	ltibia	lfemur
ikHandle 5	lfoot	ltibia
ikHandle 6	ltoes	lfoot
ikHandle 7	ltoes_end	ltoes

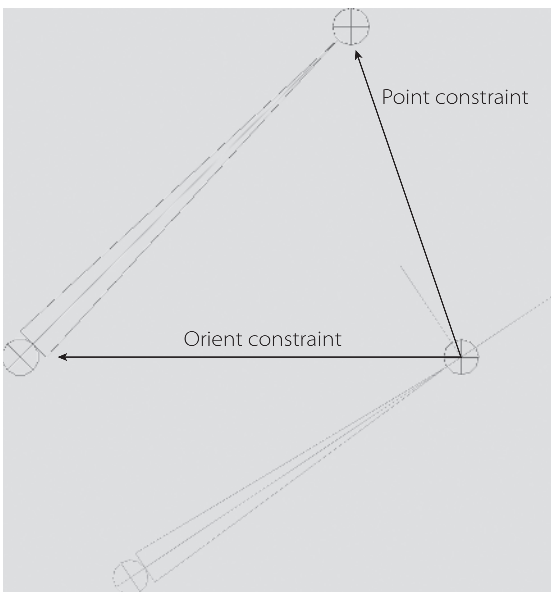


Figure 1.25
Two constraints are used to align a single bone.

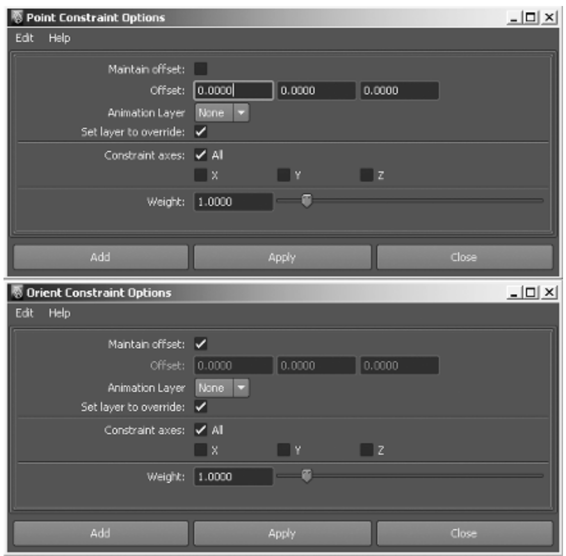


Figure 1.26
Point and orient constraint options

22. Now is a good time to check and save your work. Jack's skeleton should look like Figure 1.27 at frame 280.

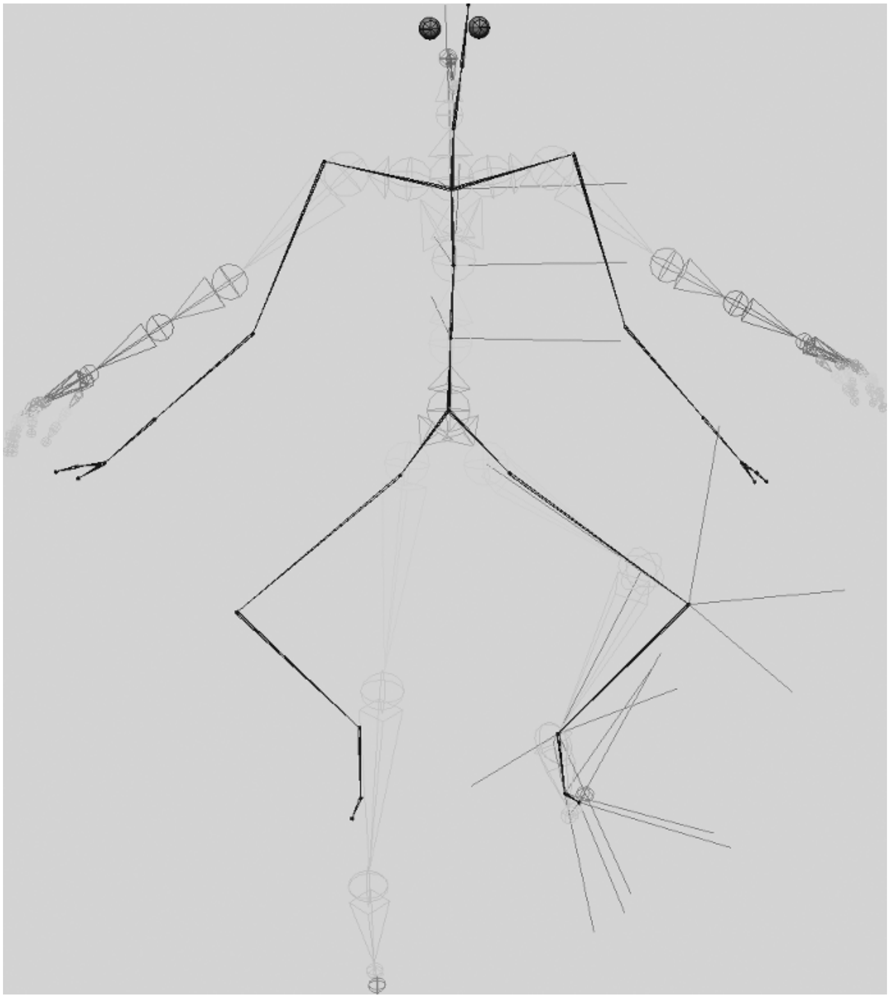


Figure 1.27
Check the alignment at frame 280.

To check your work so far, you can compare it to `mocap_v03.ma` on the DVD.

23. The left arm is next. Use the following table to map it. For the hand, add an SC IK handle from wristLft_JNT to middle1Lft_JNT. The thumb on the motion-capture skeleton is too awkward to include; you can skip that joint.

SC IK Handle	Point Constraint	Orient Constraint
ikHandle 8	lhumerus	lclavicle
ikHandle 9	lradius	lhumerus
ikHandle 10	lwrist	lradius
ikHandle 11	lhand	lwrist

To check the alignment of the two skeletons, go to frame 424 and compare it with Figure 1.28.

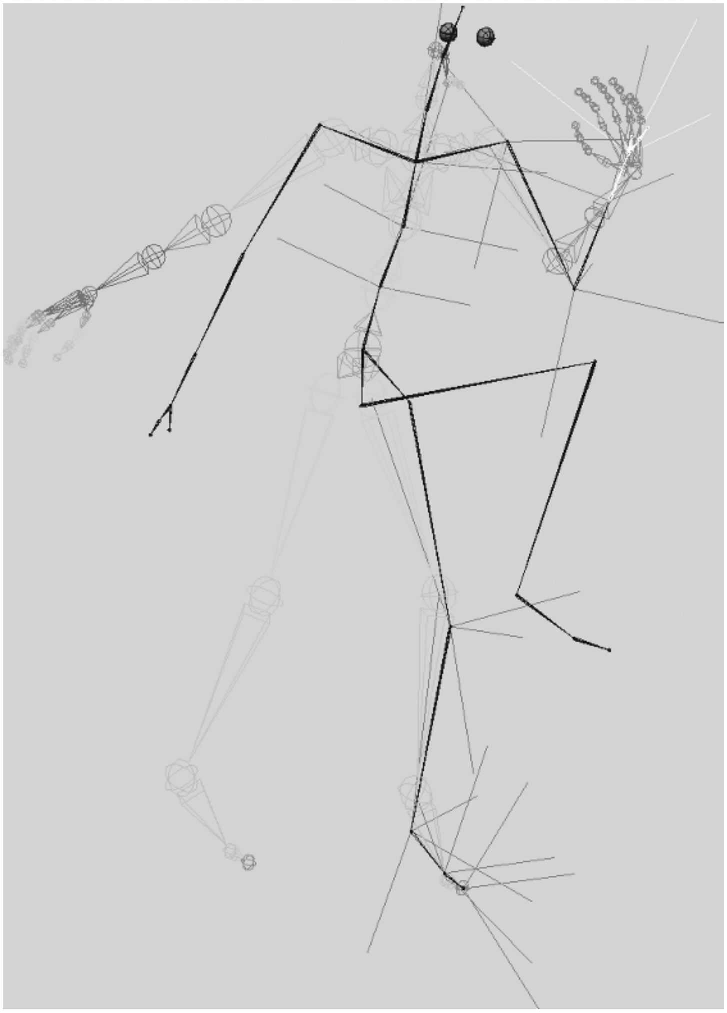


Figure 1.28
Check the alignment at frame 424.

To check your work so far, you can compare it to `mocap_v04.ma` on the DVD.

24. The head and neck are next.

SC IK Handle	**Point Constraint**	**Orient Constraint**
ikHandle 12	upperneck	lowerneck
ikHandle 13	head	upperneck

25. More than half of the character is constrained. Unfortunately, IK handles and constraints can't be mirrored. You must manually assign all the bones from the right leg and arm. You can continue or skip ahead to the following saved scene file.

To check your work so far, you can compare it to `mocap_v05.ma` on the DVD.

26. Hide all the joints and handles. Play the animation to check the quality of the motion. Figure 1.29 shows Jack in an extreme pose at frame 550.

Maintaining Performance

Other than the obvious geometry collisions, the data transferred pretty well. The only major motion-related issue is in the shoulders: they look too high. Your first thought might be to adjust the animation or offset the IK handle. Both those solutions can cause major headaches. Adjusting the IK handle only throws the timing off for the rest of the arm. Adding keyframes over the top via a layer or secondary rig has the potential to do the same thing. More important, doing so changes the performance.

Your goal is to use the data as is. It's the performance you want. Altering it, even a little, destroys the actor's intentions and, worse, the timing of the captured motion. Maintaining its integrity is paramount. In the next project, you'll address the shoulder issue without altering the performance.

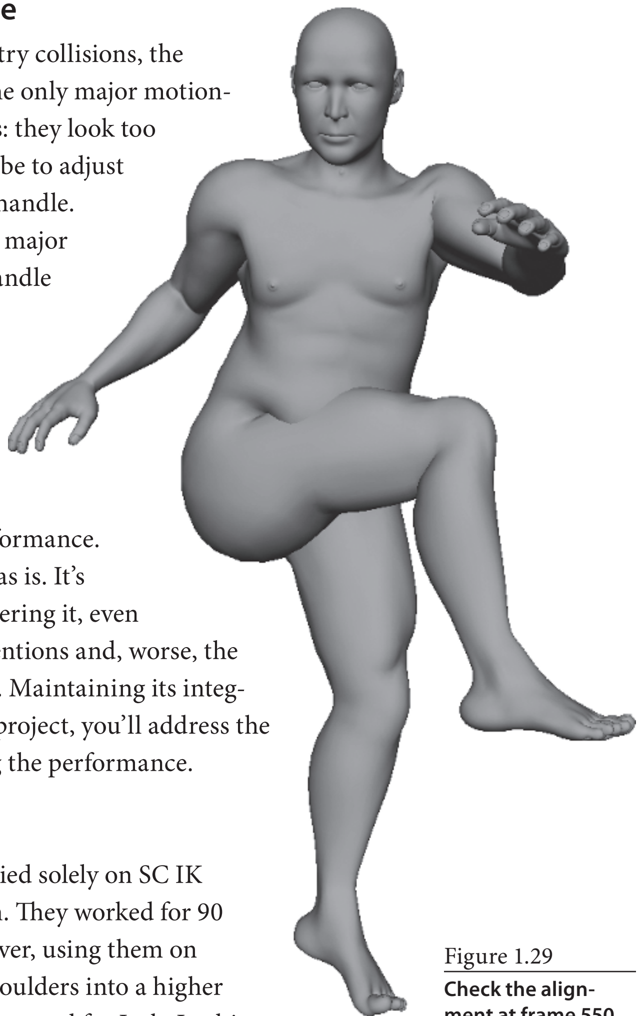


Figure 1.29
Check the alignment at frame 550.

Project: Shoulder Fix

In the previous project, you relied solely on SC IK handles to drive Jack's skeleton. They worked for 90 percent of the character. However, using them on the shoulders has forced the shoulders into a higher constrained position that isn't natural for Jack. In this project, you'll remove the handle and use only an orient constraint:

1. Open the `mocap_v05.ma` scene from the `chapter1\scenes` folder on the DVD. It's the same scene that "Project: Motion Test" ended with.
2. Go to frame 1, and look at the shoulders (see Figure 1.30).
3. Select ikHandle8, and delete it.

4. Select clavicleLft_JNT. Change all of its Rotation values to 0.0.

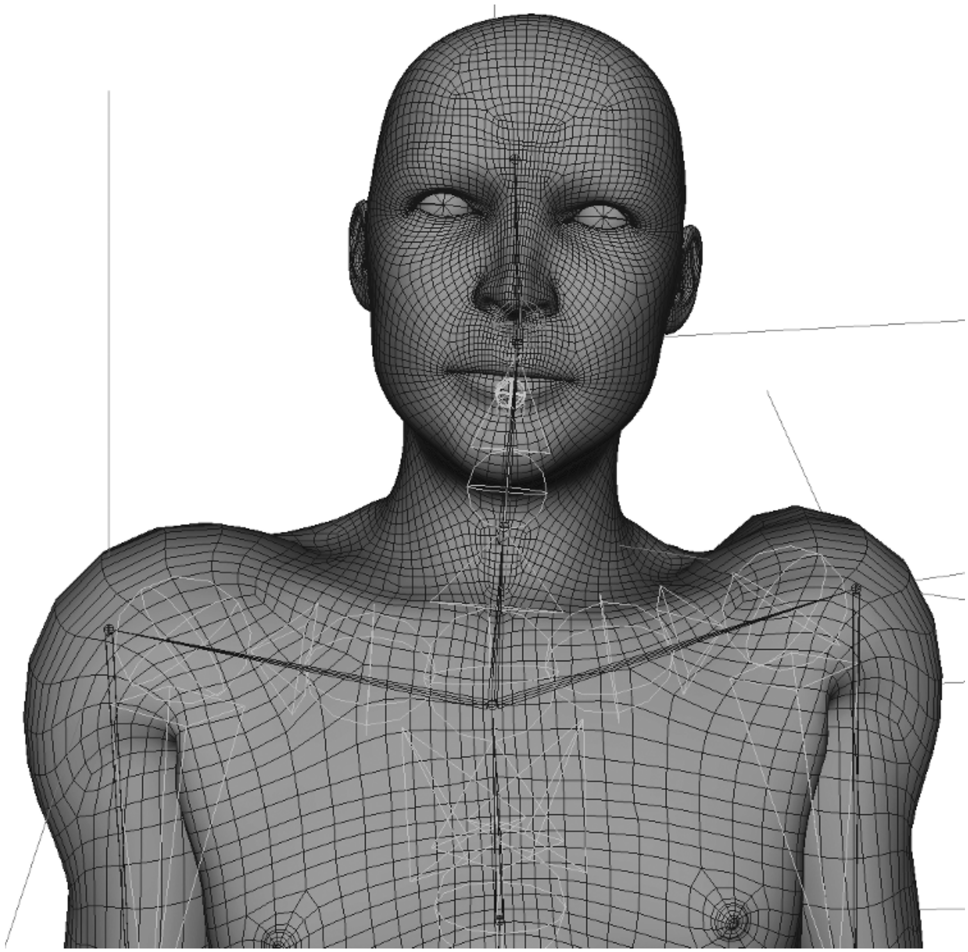


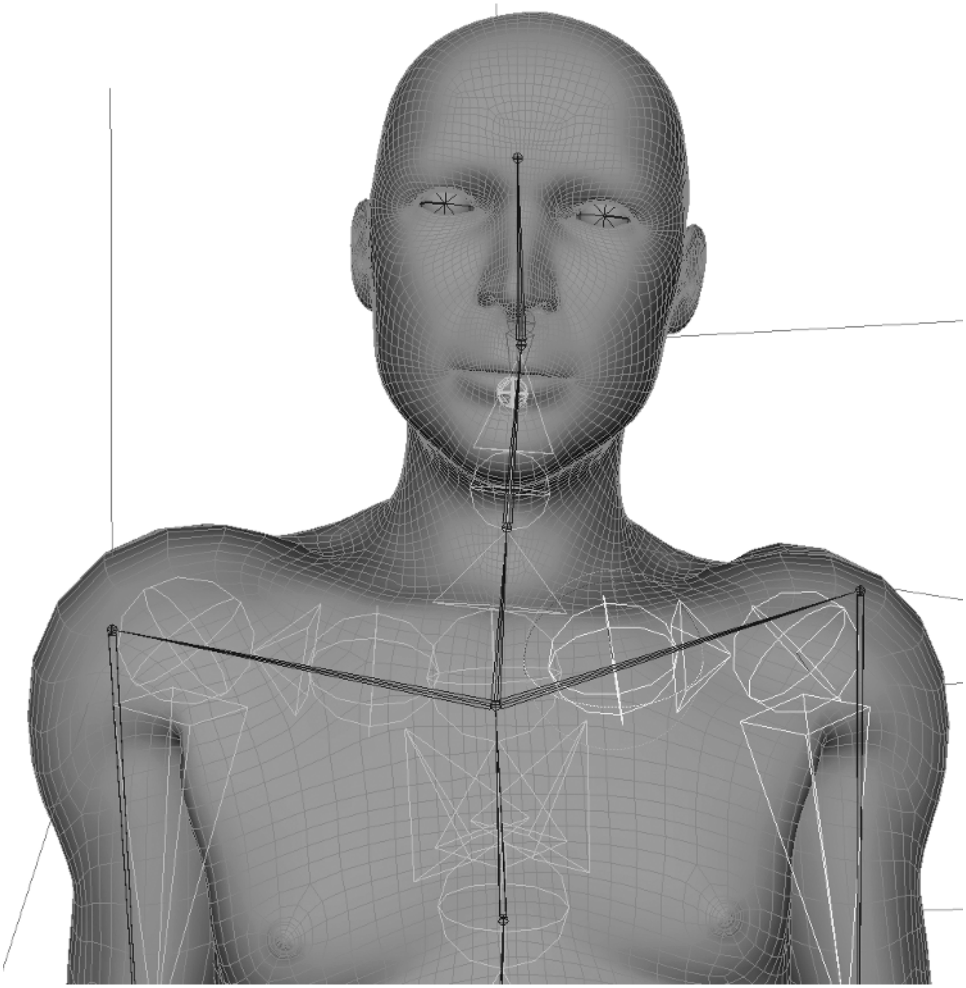
Figure 1.30
The shoulders at frame 1

UPDATING CONSTRAINTS

When you move the clavicle, the arm moves away from the motion-capture skeleton it's constrained to. The constraints aren't broken; they simply don't update. To force the constraints to update, move to a different frame and then move back to frame 1; the arm will now be in the proper position.

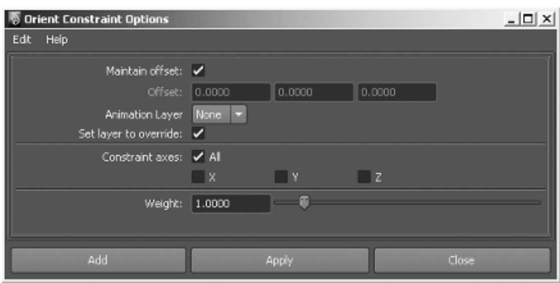
5. Even without a rotation on the clavicle, it's too high based on the position of the arm. Rotate it to 7.0 units in the *Z*. You can see the results in Figure 1.31.

Figure 1.31
The shoulder has been rotated in the *Z* by 7.0 units.



6. Shift+select lclavicle and clavicleLft_JNT. Choose Constrain → Orient. Add the constraint with the settings from Figure 1.32.

Figure 1.32
Orient constraint options



7. After evaluating the shoulder again with its new orient constraint, it becomes evident that there are weighting issues compounding the problem. You can see in Figure 1.33 that clavicleLft_JNT is too strong of an influence.

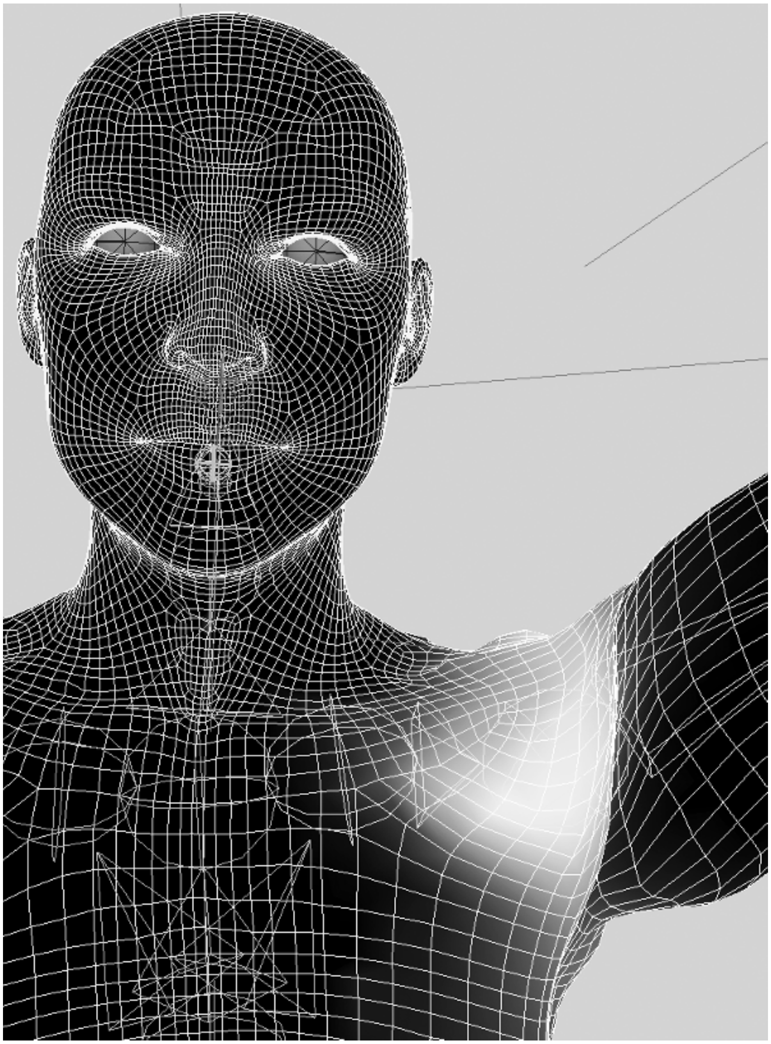


Figure 1.33
ClavicleLft_JNT weights

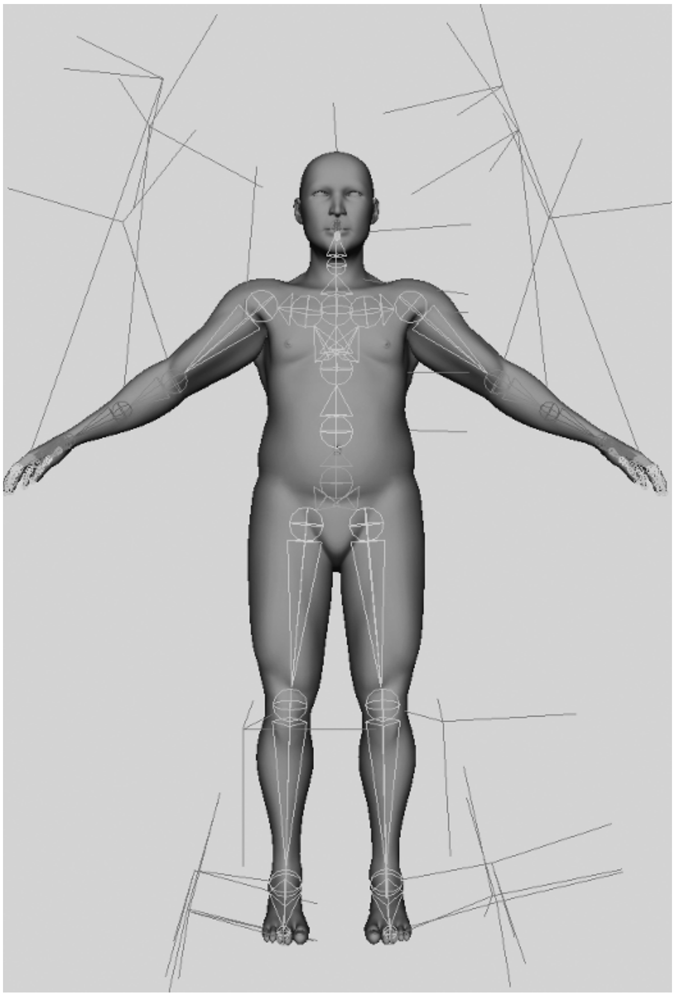
8. You don't want to paint the weights on this model. Doing so would break the smooth blending between the head and body. Instead, return to the single-mesh weighted character to perform the fix. Before proceeding, save your scene file.

   To check your work so far, you can compare it to `shoulderFix_v01.ma` on the DVD. 

9. You can work on painting the weights on the single-mesh character or import the already-fixed scene file, `weightFix_v01.ma`, from the `chapter1\scenes` folder on the DVD. 

10. The correctly weighted mesh is loaded into the scene. The two versions of Jack overlap each other. Turn off the layer visibility of the imported Jack nodes.
11. You can't copy the weights over to the animated Jack successfully until he is back into his bind pose. Go to frame -13.0 on the timeline, and choose Modify → Evaluate Nodes → Ignore All.
12. Now that everything is shut down, select all of the animated Jack's joints. A quick way to accomplish this is to choose Edit → Select All by Type → Joints. You can then deselect the extra joints in the Hypergraph or Outliner.
13. With all of Jack's joints selected, set their Rotation to 0.0 in the Channel Box. Jack returns to his default pose without all of his IK handles (see Figure 1.34).

Figure 1.34
Jack is now back in his default bind pose.



14. You can turn on the layer's visibility for the correctly weighted Jack, although doing so isn't necessary. Shift+select weightFix_v01_body1 and body1.
15. Using the default settings, choose Skin → Edit Smooth Skin → Copy Skin Weights.
16. Shift+select weightFix_v01_head1 and head1. Choose Skin → Edit Smooth Skin → Copy Skin Weights.
17. Turn all the nodes back on by choosing Modify → Evaluate Nodes → Evaluate All. Jack snaps back into place with new shoulder weights (see Figure 1.35).

To check your work so far, you can compare it to `shoulderFix_v02.ma` on the DVD.

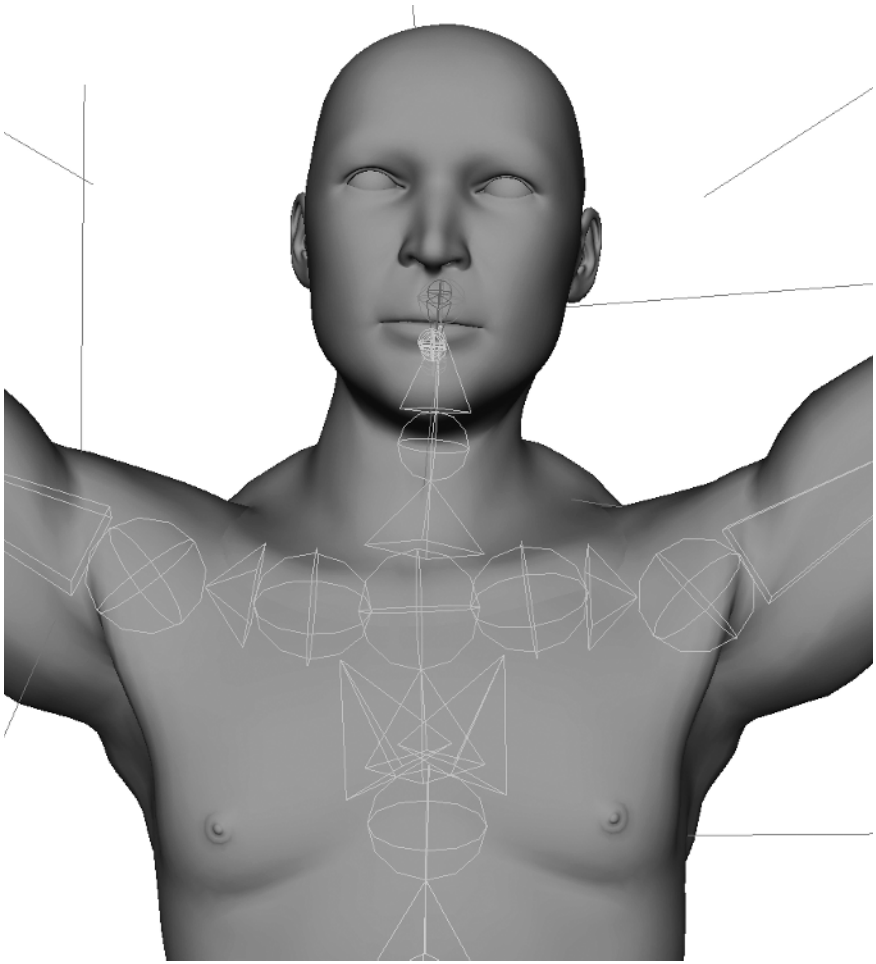


Figure 1.35
Jack looks better with his new weighting.

FBX

The FBX file format has a long history of exchanging hands. It's ironic, because it's labeled as a universal data-exchange format. The name is from its original owner, Filmbox. After numerous ups and downs, Filmbox eventually became MotionBuilder, now owned by Autodesk. Its current version is now supported by Maya, 3D Studio Max, Softimage, LightWave, and others, including a QuickTime format.

Project: FBX Import

In this project, you'll use the FBX file format to export and import a motion onto Jack. Here are the steps:

1. Open the `boxing_v01.ma` scene from the `chapter1\scenes` folder on the DVD. The scene file contains a skeleton with boxing motion. The timeline has been set to the beginning and end of the motion. The skeleton was derived from the same actor who provided the motion for the exercise routine in "Project: Motion Test." You can see the skeleton taking a punch in Figure 1.36.

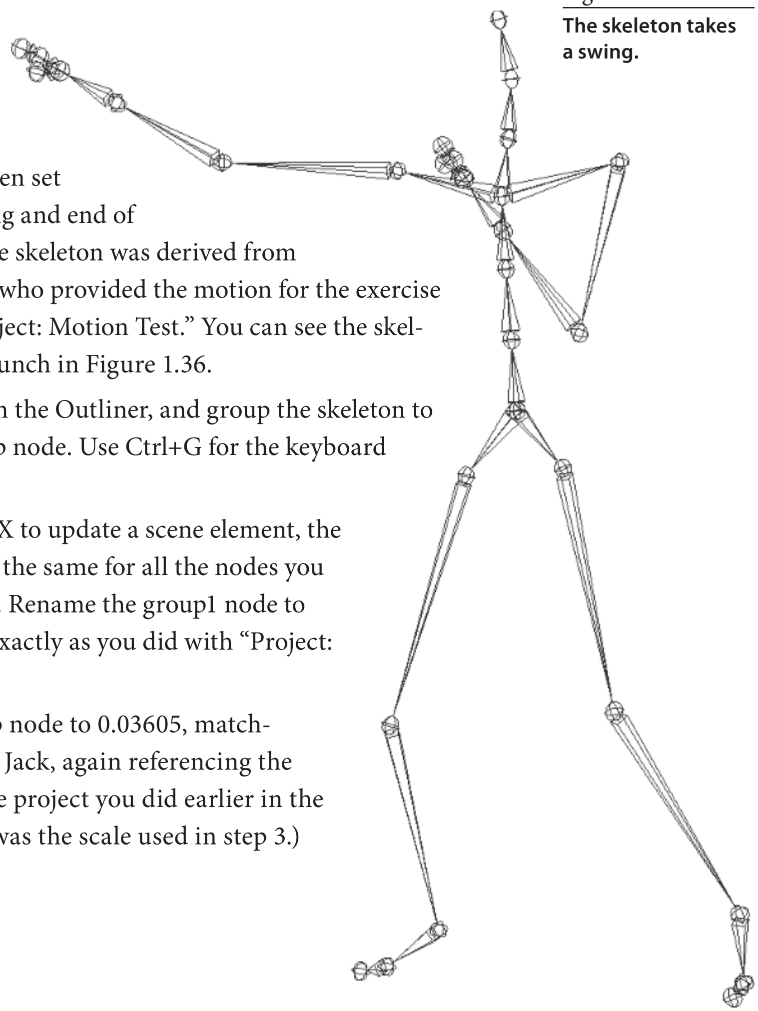


Figure 1.36

The skeleton takes a swing.

2. Select root from the Outliner, and group the skeleton to an empty group node. Use Ctrl+G for the keyboard shortcut.
3. In order for FBX to update a scene element, the names must be the same for all the nodes you want to update. Rename the group1 node to **mocap_GRP**, exactly as you did with "Project: Motion Test."
4. Scale the group node to 0.03605, matching the scale of Jack, again referencing the motion-capture project you did earlier in the chapter. (This was the scale used in step 3.)

5. Select mocap_GRP, and snap its pivot point to the root joint. You can hold D and V on the keyboard to do both actions simultaneously. This is a good time to save your scene.

   To check your work so far, you can compare it to `boxing_v02.ma` on the DVD.

6. Choose File → Export All. Change File Type to FBX Export, and click Edit Preset (see Figure 1.37).

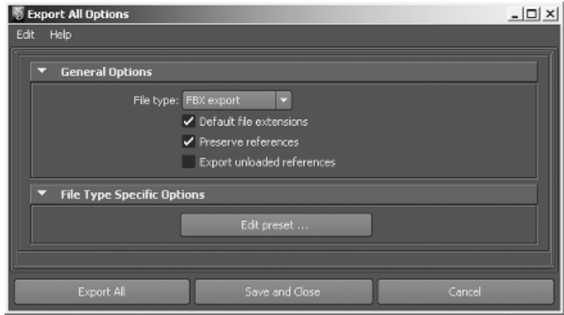


Figure 1.37
Set the Export All options to export FBX.

7. When you click Edit Preset, the Edit Export Preset window pops up. The FBX file format has numerous settings you can adjust. You only need to concern yourself with the options related to animation. We'll go through each section, ignoring the ones that are irrelevant to this export.

8. Open the Animation section: it has seven subsections. Make sure Animation is selected. Under Extra Options, make sure Use Scene Name is unchecked. You don't want to alter the name of anything in the scene; doing so would prevent the identical nodes from being updated (see Figure 1.38).

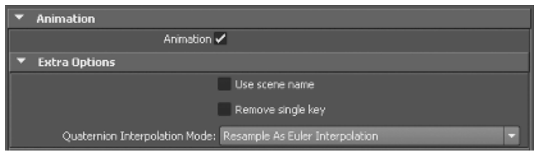


Figure 1.38
The Extra Options section of the FBX options

9. Select Bake Animation. The Start and End times should be filled in already. If they aren't, check the scene and enter the appropriate numbers (see Figure 1.39).

   Resample All is optional. Selecting it redoes all the keyframes. Ultimately, you aren't using any of the keyframes on this skeleton; you're only copying its motion, so the keyframe positions don't matter. But if you like or possibly foresee the need to look at the animation curves, this option is very useful. The original boxing motion was captured at 120 fps and reduced in Maya to playback at 30 fps. This forces the keys into fractional frame numbers. Resampling the animation through the FBX export puts a keyframe only on integer frame numbers.

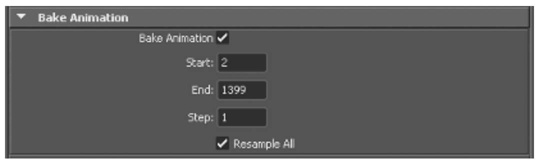


Figure 1.39
The Bake Animation section of the FBX options

10. You can deselect Deformed Models in the Deformed Models section. The parameter doesn't apply because there isn't any geometry in the scene (see Figure 1.40).

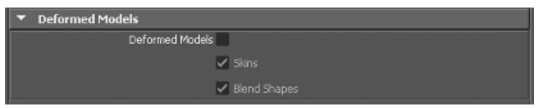
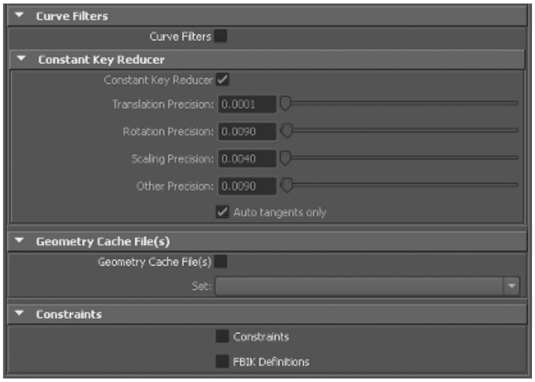


Figure 1.40
The Deformed Models section of the FBX options

11. The rest of the Animation options should be turned off. Curve Filters and Constant Key Reducer are useful for cleaning up animation curves, but at this point you aren't interested in cleaning the curves.

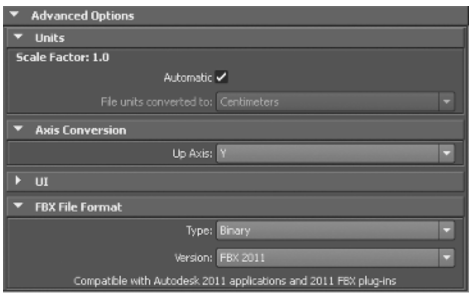
    Geometry Cache Files and Constraints, the next two sections, have no bearing on the scene because none of these objects exist in the scene. Figure 1.41 shows all the settings.

Figure 1.41
The remaining Animation options



12. The only other parameters that need mentioning are under Advanced Options: the Units and Axis Conversion sections, shown in Figure 1.42.

Figure 1.42
The Units and Axis Conversion sections of the FBX options



    None of these need altering, but they're important if you're exporting to a different software package or want to convert the units through the export. You can also set the file format to ASCII, giving you the ability to edit the output by hand.

13. Click Save Preset, and call it **mocap**. You can call the preset anytime you need to export another motion.
14. Click Close to exit the Edit Export Preset window. The Export All Options window is still open. Click Export All to save the file. It's a good idea to create a directory named `FBX` in your project directory to keep things organized.
15. Open the `shoulderFix_v02.ma` scene from the `chapter1\scenes` folder on the DVD.

16. Choose File → Import, and change File Type to FBX. Also deselect Use Namespaces, and use the settings shown in Figure 1.43.

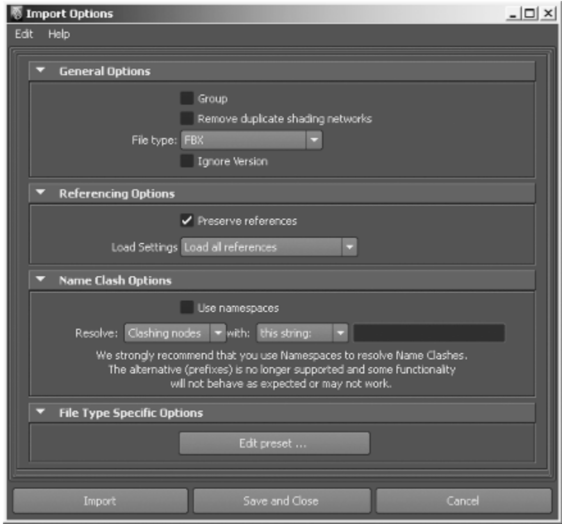


Figure 1.43
The Import Options window

17. Click Import to go to the Import browser window. Select the `boxing.fbx` scene from the `chapter1\FBX` folder on the DVD. As soon as you select the file, the FBX options appear at right in the window.
18. Under Include, change File Contents to Update Scene Elements. Make sure Animation is selected, and use the settings from Figure 1.44.

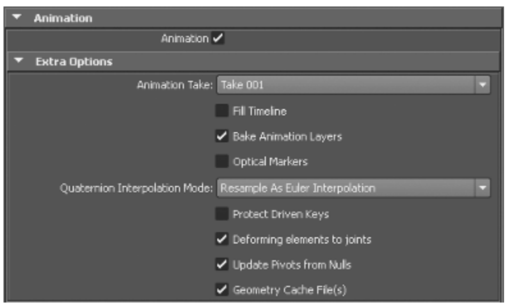


Figure 1.44
The Animation Import Options section

19. Click Import in the browser.
20. The motion loads, and the Jack character is updated. He now moves with the boxing motion capture. Figure 1.45 shows him in action at frame 87.

To check your work so far, you can compare it to `boxing_v03.ma` on the DVD.

For free and instant access motion capture, the results are pretty good. If you were to make extensive use of a motion-capture library, you could start with one of the skeletons provided by the library and build your character around it.

Summary

Libraries are limiting. At some point, you'll need an animation they don't have or some type of transition between captures. MotionBuilder by Autodesk is ideal for handling this type of work, but it adds more to your pipeline. In an ideal world, you'd have your own motion-capture studio! Keep reading, and you'll get there.

Figure 1.45
Jack taking a swing